Spring 3-1-1975

Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems ; CU-CS-063-75

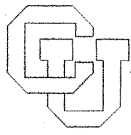
James R. Bunch
University of California at San Diego

Linda Kaufman
University of Colorado Boulder

Some Stable Methods for Calculating Inertia And Solving Symmetric Linear Systems

James R. Bunch
Linda Kaufman

CU-CS-063-75



University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems

by
James R. Bunch
Department of Mathematics
University of California at San Diego
La Jolla, California 92037

and
Linda Kaufman
Department of Computer Science
University of Colorado
Boulder, Colorado 80302

Report #CU-CS-063-75 March 1975

Abstract.

Several decompositions of symmetric matrices for calculating inertia and solving systems of linear equations are discussed. New partial pivoting strategies for decomposing symmetric matrices are introduced and analyzed.

1. Introduction.

An $n \times n$ real matrix A is symmetric if $A_{ij} = A_{ji}$ for $1 \le i,j \le n$. There are several decompositions of symmetric matrices, e.g. symmetric triangular factorization (the LDL^t decomposition) [9], the Cholesky decomposition [9], the diagonal pivoting decomposition [2, 3, 4], and the orthogonal decomposition [9]. The decomposition used depends on the problem to be solved, e.g. solving systems of linear equations, calculating inertia, or finding eigensystems.

All the statements in this paper concerning real symmetric matrices also hold for complex Hermitian matrices ($A = A^* \equiv \overline{A}^t$) by replacing t with $*$ throughout.

When solving systems of linear equations where the coefficient matrix A is nonsingular and symmetric, we may always neglect the symmetry of A and use Gaussian elimination (triangular factorization). This requires $\frac{1}{3}n^3$ multiplications, $\frac{1}{3}n^3$ additions, $\le \frac{1}{2}n^2$ comparisons, and n^2+n storage to obtain the triangular factorization of a permutation of A, i.e. $PA = LU$ where L is unit lower triangular, U is upper triangular, and P is a permutation matrix. Thus, if we want to solve $Ax = b$, we solve $Ly = Pb$ for y and then $Ux = y$ for x, each requiring $\frac{1}{2}n^2$ multiplications and $\frac{1}{2}n^2$ additions.

Can we take advantage of the symmetry of A to solve $Ax = b$ in $\frac{1}{6}n^3$ multiplications and $\frac{1}{6}n^3$ additions?

If $A = LU$ exists when A is symmetric, then $U = DL^t$, where D is diagonal (i.e. $D_{ij} = 0$ for $i \neq j$), and $A = LDL^t$ can be computed with $\frac{1}{6}n^3$ multiplications, $\frac{1}{6}n^3$ additions, and $\frac{1}{2}n^2$ storage. However, $A = LDL^t$ need not exist, e.g. $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$. In fact, $PAP^t = LDL^t$ need not exist for any permutation matrix P, e.g. $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

But if A is also positive definite or negative definite ($x^tAx > 0$, or $x^tAx < 0$, respectively, for all $x \neq 0$), then the LDL^t decomposition of A exists. If A is positive definite, then $D_{ii} > 0$ for each i, and $A = \tilde{L}\tilde{L}^t$ where $\tilde{L} = LD^{1/2}$ and $D^{1/2} = \text{diag}\{\sqrt{D_{11}}, \ldots, \sqrt{D_{nn}}\}$; this is the Cholesky decomposition.

If A is symmetric indefinite (there exist $x, y \neq 0$ such that $x^tAx > 0$ and $y^tAy < 0$), then these methods can fail (and can be unstable in finite precision arithmetic [4, pp. 643-645]).

In the eigensystem problem one seeks to find (all or some of) the eigenvalues (or, all or some of the eigenvalues and corresponding eigenvectors) of a matrix. We say that λ is an <u>eigenvalue</u> of A and $x \neq 0$ is an <u>eigenvector</u> corresponding to λ if $Ax = \lambda x$.

Since an $n \times n$ symmetric matrix A has only real eigenvalues [9], let u, v, w be the number of positive, negative, zero eigenvalues, respectively. The triple (u, v, w) is called the <u>inertia</u> of A, while $s \equiv u - v$ is called the <u>signature</u> of A. But $n = u + v + w$ is the <u>order</u>

of A and $r \equiv u+v$ is the rank of A. Thus, $u = \frac{1}{2}(r+s)$, $v = \frac{1}{2}(r-s)$, and $w = n - r$. Knowing the order, rank, and signature of a symmetric matrix A is equivalent to knowing the inertia of A. If A is non-singular, then $w = 0$ and $r = n$, so knowing the inertia is equivalent to knowing the signature. Note that in the inertia problem we are seeking only the signs of the eigenvalues, not the eigenvalues themselves, and hence we seek some method that would be faster than calculating all the eigenvalues (cf. Cottle [6]).

Suppose $A = LDL^t$, where L is unit lower triangular and D is diagonal. We shall show below that u, v, w are equal to the number of positive, negative, zero elements, respectively on the diagonal of D. Since it requires only $\frac{1}{6}n^3$ multiplications to compute the LDL^t decomposition, this is much less work than calculating the eigenvalues.

Unfortunately, the LDL^t decomposition of a symmetric matrix need not exist if A is indefinite. If A is positive (negative) definite, then the LDL^t decomposition of A always exists and the eigenvalues of A are all positive (negative) since $Ax = \lambda x$ for $x \neq 0$ implies $\lambda = \frac{x^tAx}{x^tx} > 0$ (< 0); so $u = n = r = s$ and $v = 0 = w$ $(v = n = r = s$ and $u = 0 = w)$.

The theoretical foundation for calculating inertia is provided by Sylvester's Inertia Theorem [7, pp. 371-372]; the inertia of a symmetric

matrix is invariant under nonsingular congruences, i.e. if A is symmetric and S is nonsingular, then A and SAS^t have the same inertia, and hence the same rank and signature.

The classical method for calculating the inertia of a symmetric matrix is based on Lagrange's method for the reduction of a quadratic form to a diagonal form.

If A is an $n \times n$ matrix and x is an n-vector, then we say that $\varphi(x) \equiv x^t A x = \sum_{i,j=1}^{n} A_{ij} x_i x_j$ is a <u>quadratic form</u>. If A is of rank r, then we say that $\varphi(x)$ is a quadratic form of rank r.

Note that $B \equiv \frac{1}{2}(A + A^t)$ is symmetric and $x^t B x = x^t A x$. Hence, without loss of generality, we may assume that A is symmetric.

Lagrange's method (1759) reduces a quadratic form $\varphi(x) = x^t A x$ to a diagonal form $z^t D z$, where D is a diagonal matrix with exactly $r = \text{rank}(A)$ nonzero elements, i.e. $x^t A x = z^t D z$ where $z = Sx$, $\det S \neq 0$. Since $A = S^t D S$, by Sylvester's Inertia Theorem, A and D have the same inertia.

Let us consider Lagrange's method in detail. If $A_{11} \neq 0$, then

$$\varphi(x) = x^t A x = \sum_{i,j=1}^{n} A_{ij} x_i x_j =$$

$$A_{11} x_1^2 + 2A_{12} x_1 x_2 + \ldots + 2A_{1n} x_1 x_n + \sum_{i,j=2}^{n} A_{ij} x_i x_j =$$

$$A_{11}\left(x_1^2 + 2\frac{A_{12}}{A_{11}} x_1 x_2 + \ldots + 2\frac{A_{1n}}{A_{11}} x_1 x_n\right) + \sum_{i,j=2}^{n} A_{ij} x_i x_j =$$

$$A_{11}\left(x_1 + \frac{A_{12}}{A_{11}} x_2 + \ldots + \frac{A_{1n}}{A_{11}} x_n\right)^2 + \sum_{i,j=2}^{n} \left(A_{ij} - \frac{A_{1i} A_{1j}}{A_{11}}\right) x_i x_j .$$

Thus, take $D_{11} = A_{11}$ and $z_1 = x_1 + \frac{A_{12}}{A_{11}} x_2 + \ldots + \frac{A_{1n}}{A_{11}} x_n$. This is also called the method of completing the square.

Note that this is identical to the first step of the LDL^t decomposition of A. If $A_{11} \neq 0$, then $A =$

$$\left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline \frac{A_{21}}{A_{11}} & 1 & & 0 \\ \vdots & & \ddots & \\ \frac{A_{n1}}{A_{11}} & 0 & & 1 \end{array}\right] \left[\begin{array}{c|ccc} A_{11} & 0 & \cdots & 0 \\ \hline 0 & A_{ij} - \frac{A_{i1}A_{1j}}{A_{11}} & & \\ \vdots & & & \\ 0 & & & \end{array}\right] \left[\begin{array}{c|ccc} 1 & \frac{A_{12}}{A_{11}} & \cdots & \frac{A_{1n}}{A_{11}} \\ \hline 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & 1 \end{array}\right] .$$

If $A_{22} - \frac{A_{21}^2}{A_{11}} \neq 0$, we may continue with the process.

If $A_{11} = 0$, but $A_{kk} \neq 0$ for some k, then we may interchange the $k^{\underline{th}}$ and first rows and columns and proceed as before, i.e.

$$\varphi(x) = A_{kk}\left(x_k + \frac{A_{k2}}{A_{kk}} x_2 + \cdots + \frac{A_{k,k-1}}{A_{kk}} x_{k-1} + \frac{A_{k1}}{A_{kk}} x_1 + \right.$$

$$\left. + \frac{A_{k,k+1}}{A_{kk}} x_{k+1} + \cdots + \frac{A_{kn}}{A_{kk}} x_n\right)^2 + \sum_{\substack{i,j=1 \\ i,j\neq k}} \left(A_{ij} - \frac{A_{ki}A_{kj}}{A_{kk}}\right) x_i x_j .$$

Here, take $D_{11} = A_{kk}$ and $z_1 = x_k + \frac{A_{k2}}{A_{kk}} x_2 + \cdots + \frac{A_{k,k-1}}{A_{kk}} x_{k-1} + \frac{A_{k1}}{A_{kk}} x_1 + \frac{A_{k,k+1}}{A_{kk}} x_{k+1} + \cdots + \frac{A_{kn}}{A_{kk}} x_n$.

In matrix form, let P be the permutation matrix obtained from interchanging the $k^{\underline{th}}$ and first columns of the $n \times n$ identity matrix. Then $\varphi(x) = x^t Ax = x^t P^t PAP^t Px = (Px)^t (PAP^t)(Px)$, where $Px = [x_k, x_2, \cdots, x_{k-1}, x_1, x_{k+1}, \cdots, x_n]^t$ and $(PAP^t)_{11} = A_{kk}$. Thus, $PAP^t =$

$$\begin{bmatrix} 1 & 0 \\ \hline \frac{A_{2k}}{A_{kk}} & \multicolumn{1}{|c}{} \\ \vdots & \multicolumn{1}{|c}{\begin{matrix} 1 & 0 \\ & \ddots \\ 0 & & 1 \end{matrix}} \\ \frac{A_{1k}}{A_{kk}} & \multicolumn{1}{|c}{} \\ \vdots & \multicolumn{1}{|c}{} \\ \frac{A_{nk}}{A_{kk}} & \multicolumn{1}{|c}{} \end{bmatrix} \begin{bmatrix} A_{kk} & 0 \\ \hline 0 & A_{ij} - \frac{A_{ik}A_{kj}}{A_{kk}} \end{bmatrix} \begin{bmatrix} 1 & \frac{A_{k2}}{A_{kk}} \cdots \frac{A_{k1}}{A_{kk}} \cdots \frac{A_{kn}}{A_{kk}} \\ \hline 0 & \multicolumn{1}{|c}{\begin{matrix} 1 & 0 \\ & \ddots \\ 0 & & 1 \end{matrix}} \end{bmatrix}$$

But, what do we do if $A_{11} = \cdots = A_{nn} = 0$ (or if at some stage of the process the diagonal of the remaining submatrix is all zero)? If $A \equiv 0$ then the rank of A is zero and we take $D \equiv 0$ and $z = x$. If $A \not\equiv 0$ but $A_{11} = \cdots = A_{nn} = 0$, then $A_{rs} \neq 0$ for some $r \neq s$. We can now interchange the $r^{\underline{th}}$ and first, and the $s^{\underline{th}}$ and second, rows and columns, and then the $(2, 1)$ element of the resulting matrix is nonzero. Without loss of generality, we may assume $A_{21} \neq 0$.

Let $y_1 = \frac{1}{2}\left(x_1 + x_2\right)$, $y_2 = \frac{1}{2}\left(x_1 - x_2\right)$, and $y_i = x_i$ for $3 \le i \le n$, i.e. $y = T^{-1}x$, where $T = S \oplus I_{n-2}$, $S = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and I_{n-2} is the identity matrix of order $n - 2$. Then $x^tAx = y^t(T^tAT)y$ with $(T^tAT)_{11} = 2A_{12} \neq 0$. Thus we may proceed as before.

Let us explore the above in matrix form. Let $A = \begin{bmatrix} E & C^t \\ C & B \end{bmatrix}$, $T = \begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix}$, $\det E \neq 0$, order (E) = order (S). Then $T^tAT = \begin{bmatrix} S^tES & S^tC^t \\ CS & B \end{bmatrix}$. Lagrange chose S to be of order 2 so that $S^tES = D$ was diagonal. If we used E as a block pivot in A and performed a block decomposition, then the reduced matrix is $B - CE^{-1}C^t = B - CSD^{-1}S^tC^t$. Thus we need not find a 2×2 matrix S which diagonalizes E, but we can use E itself and do a block decomposition with the 2×2 submatrix E. If the diagonal of A is null then there exists a nonsingular principal 2×2 submatrix unless $A \equiv 0$.

Hence, given any symmetric matrix A, there exists a permutation matrix P such that

$$PAP^t = MDM^t , \tag{1.1}$$

where M is unit lower triangular, D is block diagonal with blocks of order 1 or 2, and $M_{i+1,i} = 0$ whenever $D_{i+1,i} \neq 0$.

Let us look at the determinant of such a block of order 2 :

$$\det \begin{bmatrix} 0 & D_{i,i+1} \\ D_{i+1,i} & 0 \end{bmatrix} = - D_{i+1,i} D_{i,i+1} = - D_{i+1,i}^2 < 0 .$$

Hence, by Sylvester's Inertia Theorem, one positive and one negative eigenvalue of A is associated with this 2×2 block.

Let p be the number of 1×1 blocks in D. Hence there are $q = \frac{1}{2}(n - p)$ blocks of order 2. Let a, b, c be the number of positive, negative, and zero 1×1 blocks. Thus the inertia of A is $(a+q, b+q, c)$, the rank of A is $n - c$, and the signature of A is $a - b$.

In finite precision arithmetic on a computer, in order to maintain stability and insure a good solution we must prevent large growth in the elements of the reduced matrices generated during the decomposition process [4, 9] . Hence, we will want to use 2×2 pivots whenever the diagonal is small as well as whenever the diagonal is null [4, 9] . Our knowledge of the inertia will be preserved as long as the determinant of each 2×2 pivot remains negative.

Based on the above method, called the diagonal pivoting method, Bunch [2] showed that inertia can be calculated and nonsingular symmetric systems of linear equations can be solved by only $\frac{1}{6} n^3$ multiplications, $\frac{1}{6} n^3$ additions, and $\frac{1}{2} n^2$ storage. The method is almost as stable as Gaussian

elimination with complete pivoting. The price paid for stability is $\geq \frac{1}{12} n^3$ but $\leq \frac{1}{6} n^3$ comparisons.

Let us consider the first step of the algorithm. Let $A = \begin{bmatrix} E & C^t \\ C & B \end{bmatrix}$ where E is of order $s = 1$ or 2. Let $\mu_0 = \max_{i,j} |A_{ij}|$, $\mu_1 = \max_{i} |A_{ii}|$, $\nu = |\det E|$.

If $s = 1$, then $|E| = \nu$. Assume $\nu \neq 0$. Then

$$\begin{bmatrix} E & C^t \\ C & B \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ C/E & I_{n-1} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & A^{(n-1)} \end{bmatrix} \begin{bmatrix} 1 & C^t/E \\ 0 & I_{n-1} \end{bmatrix} ,$$

where $A^{(n-1)} = B - CC^t/E$ and I_{n-1} is the identity matrix of order n-1. So,

$$\max_{i,j} |A_{ij}^{(n-1)}| \leq \mu_0 + \mu_0^2/\nu .$$

Thus large element growth will not occur for a 1×1 pivot if $\nu = |E|$ is large relative to μ_0.

If E is of order 2 and $\nu \neq 0$, then

$$\begin{bmatrix} E & C^t \\ C & B \end{bmatrix} = \begin{bmatrix} I_2 & 0 \\ CE^{-1} & I_{n-2} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & A^{(n-2)} \end{bmatrix} \begin{bmatrix} I_2 & E^{-1}C^t \\ 0 & I_{n-2} \end{bmatrix} ,$$

where $A^{(n-2)} = B - CE^{-1}C^t$. So ,

$$\max_{i,j} |A_{ij}^{(n-2)}| \leq \left[1 + \frac{2\mu_0(\mu_0+\mu_1)}{\nu}\right]\mu_0 .$$

Let α be a fixed number with $0 < \alpha < 1$. We shall use a 1×1 pivot if $\mu_1 \geq \alpha\mu_0$. If so, we interchange the $1^{\underline{st}}$ and $k^{\underline{th}}$ row and column, where $\mu_1 = \max_i |A_{ii}| = |A_{kk}|$. Without loss of generality, we may assume $|A_{11}| = \mu_1$. Hence, $\nu = \mu_1$ and

$$\max_{i,j} |A_{ij}^{(n-1)}| \leq (1+1/\alpha)\mu_0 .$$

If $\mu_1 < \alpha\mu_0$, where $\mu_0 = |A_{rq}|$ for $r \neq q$, then interchange the $r^{\underline{th}}$ and $2^{\underline{nd}}$ row and column and the $q^{\underline{th}}$ and $1^{\underline{st}}$ row and column, and use a 2×2 pivot. Without loss of generality, we may assume $\mu_0 = |A_{21}|$. Then,

$$\nu = |A_{11}A_{22} - A_{21}^2| = \mu_0^2 - A_{11}A_{22} \geq \mu_0^2 - \mu_1^2 > (1-\alpha^2)\mu_0 .$$

and

$$\max_{i,j} |A_{ij}^{(n-2)}| \leq [1 + 2/(1-\alpha)]\mu_0 .$$

Thus, all the elements in all the reduced matrices are bounded by $B(\alpha)^{n-1}$, where $B(\alpha) = \max\{1+1/\alpha\ ,\ [1+2/(1-\alpha)]^{1/2}\}$.

Now $\min_{0<\alpha<1} B(\alpha) = B(\alpha_0) = \left(1+\sqrt{17}\right)/2 < 2.57$, where $\alpha_0 = \left(1+\sqrt{17}\right)/8 \doteq 0.6404$.

Since the largest element in each reduced matrix is calculated, this is a complete pivoting strategy analogous to Gaussian elimination with complete pivoting.

Furthermore, Bunch [2] proves that the element growth in the diagonal pivoting method with complete pivoting is bounded by $3n\ f(n)$ in comparison with $\sqrt{n}\ f(n)$ for Gaussian elimination with complete pivoting, where

$$f(n) = \left(\prod_{k=2}^{n} k^{\frac{1}{k-1}} \right)^{1/2} < 1.8\ n^{\frac{1}{4}\log n}.$$

In [3] Bunch discusses various partial pivoting strategies for the diagonal pivoting method which require only $O(n^2)$ comparisons instead of $O(n^3)$, although these increase element growth. In this paper we shall present and analyze several good partial pivoting algorithms for the diagonal pivoting method.

In § 2 we shall show that the diagonal pivoting method can be modified so that only n^2 comparisons are needed but element growth is now bounded by $(2.57)^{n-1}$ (cf. 2^{n-1} for Gaussian elimination with partial pivoting).

In § 3 other variations of the algorithm are presented and analyzed.

In § 4 the situation for symmetric indefinite band matrices is discussed. We are unable to give an algorithm which preserves the banded structure for every bandwidth $2m+1$. However, we are able to give good algorithms for the important special cases when $m=1$ and $m=2$ (tridiagonal and 5-diagonal).

2. A Partial Pivoting Strategy

In this section we describe and analyze a partial pivoting strategy for transforming an n x n symmetric indefinite matrix A by stable congruences into a block diagonal matrix D, where each block is of order 1 or 2. As in Bunch and Parlett's [4] complete pivoting strategy, the algorithm generates a sequence of matrices $A^{(k)}$ of order k according to the formula

$$A^{(k-s)} = B - CE^{-1}C^t$$

where $A^{(k)}$ has been permuted so that it can be partitioned into

$$A^{(k)} = \left[\begin{array}{c|c} E & C^t \\ \hline C & B \end{array}\right]$$

where E is an s x s nonsingular matrix, C is a (k-s)xs matrix, and B is a (k-s)x(k-s) matrix and s is either 1 or 2, depending on whether a 1 x 1 or 2 x 2 pivot is used.

Bunch and Parlett's pivoting strategy may be considered analogous to Gaussian elimination with complete pivoting. Unfortunately, there is no stable scheme exactly analogous to Gaussian elimination with partial pivoting; one cannot construct an algorithm for which there is a bound on the element growth of the sequence $A^{(k)}$ when at each stage only one column of $A^{(k)}$ is examined (see [3]). The method described in this section guarantees that the element growth in $A^{(k)}$ is bounded while searching for the largest element in at most two columns in each $A^{(k)}$. For

future reference we call the strategy algorithm A.

In algorithm A, the matrix $A^{(k-s)}$ is determined as follows:

(1) Determine $\lambda^{(k)}$, the absolute value of the largest off-diagonal element in absolute value in the first column of $A^{(k)}$, i.e.

$$\lambda^{(k)} = |A_{j1}^{(k)}| = \max_{2 \le i \le k} |A_{i1}^{(k)}| \quad .$$

(2) If $|A_{11}^{(k)}| > \alpha \lambda^{(k)}$ where $0 < \alpha < 1$, perform a 1 x 1 pivot to obtain $A^{(k-1)}$, decrease k by 1 and return to (1). We will show that a good value for α is $(1+\sqrt{17})/8$.

(3) Determine $\sigma^{(k)}$, the absolute value of the largest off-diagonal element in absolute value in the j^{th} column of $A^{(k)}$, i.e.

$$\sigma^{(k)} = \max_{\substack{1 \le m \le k \\ m \ne j}} |A_{m,j}^{(k)}| \quad .$$

(Recall that A_{j1} is the largest off-diagonal element in the first column.)

(4) If $\alpha \lambda^{(k)^2} \le |A_{11}^{(k)}| \sigma^{(k)}$, then perform a 1 x 1 pivot to obtain $A^{(k-1)}$, decrease k by 1, and return to (1). (We need this test to guarantee stability.)

(5) If $|A_{jj}^{(k)}| \ge \alpha \sigma^{(k)}$, then interchange the first and j^{th} rows and columns of $A^{(k)}$, perform a 1 x 1 pivot with the new $A^{(k)}$, decrease k by 1, and return to (1).

(6) Interchange the second and j^{th} rows and columns of $A^{(k)}$ so that $|A_{21}^{(k)}| = \lambda^{(k)}$, perform a 2 x 2 pivot to obtain $A^{(k-2)}$, decrease k by 2 and return to (1).

Step (4) of the algorithm deserves an explanation. The step was designed to screen out a pathological case with 2 x 2 pivoting when the largest off-diagonal element in absolute value of the j^{th} column was larger than that of the first column, i.e. when $\sigma^{(k)} > \lambda^{(k)}$. In this case, step (4) is equivalent to:

> scaling the first first row and column of $A^{(k)}$ so that the absolute value of the largest element in the first column of $A^{(k)}$ is equal to $\sigma^{(k)}$ and repeating steps (1) and (2) on the scaled matrix.

In the absence of roundoff error, the reduced matrix $A^{(k-s)}$ generated by algorithm A and the one generated by using explicit scaling would be the same. If a 2 x 2 pivot had been performed when the test in step (4) dictated the use of a 1 x 1 pivot, then the element growth of $A^{(k-2)}$ could not be bounded apriori. Whenever $\lambda^{(k)} \geq \sigma^{(k)}$, the test in step (4) cannot be passed and one proceeds with step (5).

Note that whenever a 2 x 2 pivot is used, $A_{11}^{(k)}A_{22}^{(k)} < \alpha\, A_{21}^{(k)2} < A_{21}^{(k)2}$ so that a 2 x 2 block in D corresponds to a positive-negative pair of eigenvalues. This means that if A is positive definite then D will be diagonal.

We shall now analyze Algorithm A. Let $\mu = \max_{1\leq i,j\leq n} |A_{ij}|$ and

$\mu^{(k)} = \max_{1\leq i,j\leq n} |A_{ij}^{(k)}|$ for each reduced matrix $A^{(k)}$ that exists

(If $A^{(k)}$ uses a 2 x 2 pivot then $A^{(k-1)}$ does not exist). Note that

both $\lambda^{(k)}$ and $\sigma^{(k)}$ are less than or equal to $\mu^{(k)}$.

If a 1 x 1 pivot is used,

$$A_{ij}^{(k-1)} = A_{i+1,j+1}^{(k)} - \frac{A_{i+1,1}^{(k)} A_{j+1,1}^{(k)}}{A_{11}^{(k)}}$$

so that by step (2) of Algorithm A

$$\mu^{(k-1)} \leq \mu^{(k)} + \lambda^{(k)}/\alpha \leq \mu^{(k)}(1+1/\alpha) , \tag{2.1}$$

by step (4),

$$\mu^{(k-1)} \leq \mu^{(k)} + \lambda^{(k)2}/|A_{11}^{(k)}| \leq \mu^{(k)} + \sigma^{(k)}/\alpha \leq \mu^{(k)}(1+1/\alpha) \tag{2.2}$$

and by step (5),

$$\mu^{(k-1)} \leq \mu^{(k)} + \sigma^{(k)}/\alpha \leq \mu^{(k)}(1 + 1/\alpha). \tag{2.3}$$

If a 2 x 2 pivot is used,

$$A_{ij}^{(k-2)} = A_{i+2,j+2}^{(k)} - [(A_{i+2,1}^{(k)} A_{22}^{(k)} - A_{i+2,2}^{(k)} A_{21}^{(k)}) A_{j+2,1}^{(k)} + (A_{i+2,2}^{(k)} A_{11}^{(k)} - A_{i+2,1}^{(k)} A_{21}^{(k)}) A_{j+2,2}^{(k)}]/(A_{11}^{(k)} A_{22}^{(k)} - A_{21}^{(k)2}). \tag{2.4}$$

Since $|A_{11}^{(k)}| \sigma^{(k)} < \alpha \lambda^{(k)2}$ by step (5) and

$|A_{22}^{(k)}| < \alpha \sigma^{(k)}$ by step (4),

$|A_{11}^{(k)}| |A_{22}^{(k)}| < \alpha^2 \lambda^{(k)2}$ which implies that

$$\nu \equiv |A_{11}^{(k)} A_{22}^{(k)} - A_{21}^{(k)}|^2 > \lambda^{(k)2}(1-\alpha^2) \text{ or } 1/\nu < 1/(\lambda^{(k)2}(1-\alpha^2)). \tag{2.5}$$

Equations (2.4) and (2.5) together imply that

$$\mu^{(k-2)} \leq \mu^{(k)} + \frac{(\lambda^{(k)}\alpha\,\sigma^{(k)} + \sigma^{(k)}\,\lambda^{(k)})\,\lambda^{(k)} + (\sigma^{(k)}|A_{11}^{(k)}| + \lambda^{(k)2})\,\sigma^{(k)}}{\lambda^{(k)2}(1-\alpha^2)}.$$

Since $\sigma^{(k)}|A_{11}^{(k)}| < \alpha\,\lambda^{(k)2}$ and $|A_{11}^{(k)}| < \alpha\,\lambda^{(k)}$,

$$\mu^{(k-2)} \leq \mu^{(k)} + (\alpha\,\sigma^{(k)} + \sigma^{(k)} + \alpha\,\sigma^{(k)} + \sigma^{(k)})/(1-\alpha^2)$$

$$\leq \mu^{(k)}\,(1 + 2(1+\alpha)/(1-\alpha^2)) = \mu^{(k)}(1 + 2/(1-\alpha)). \tag{2.6}$$

By (2.1), (2.2), (2.3), and (2.6),

$$\max_k \mu^{(k)} \leq \max\{(1+1/\alpha)^{n-k}, (1+2/(1-\alpha))^{(n-k)/2}\}\mu.$$

The growth is minimized when

$$(1+1/\alpha)^2 = (1+2/(1-\alpha)),$$

i.e. when $\alpha = (1 + \sqrt{17})/8 \doteq 0.6406$,

in which case

$$\max_k \mu^{(k)} < \mu(2.57)^{n-1}.$$

As noted above, algorithm A is equivalent to one which scales the first row and column of $A^{(k)}$ at each step so that the maximum norm of the first and second columns of $A^{(k)}$ are equal. If the scaling had been done explicitly, then the algorithm would determine a permutation matrix P, a lower triangular matrix $\overline{M}$, and a block diagonal matrix $\overline{D}$ such that

$$PAP^t = \overline{M}\overline{D}\overline{M}^t$$

where $|\overline{M}_{ij}| \leq \max(1/\alpha, 1/(1-\alpha))$

and $|D_{ij}| \leq (2.57)^{n-1}\,\mu$.

Algorithm A creates the same matrix P, but a unit lower triangular matrix M and a block diagonal matrix D such that

$$PAP^t = MDM^t$$

where $M = \bar{M}S$

and $D = S^{-1}\bar{D}S^{-1}$

where S is a diagonal matrix given by

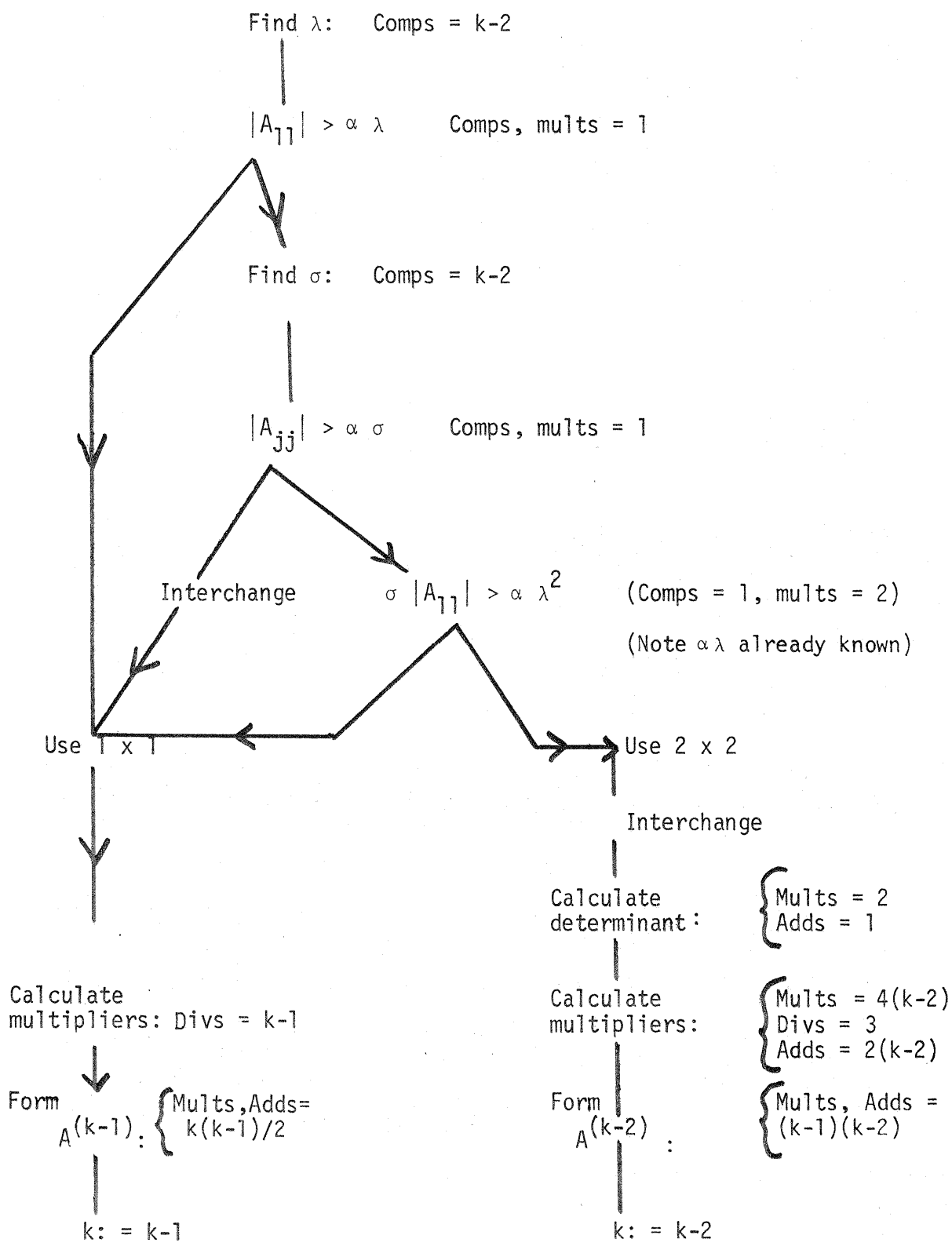
$$S_{kk} = \begin{cases} 1 \text{ if } (|A_{11}^{(k)}| \geq \alpha\lambda^{(k)}) \text{ or } (|A_{22}^{(k)}| \geq \alpha\sigma^{(k)}) \text{ or } (\lambda^{(k)} \geq \sigma^{(k)}) \\ \min\left(\sqrt{\alpha\sigma^{(k)}/|A_{11}^{(k)}|}\,,\ \sigma^{(k)}/\lambda^{(k)}\right) \text{ otherwise.} \end{cases}$$

i.e. $|S_{k,k}| \geq 1$.

The bound on the elemental growth of D is that of $\bar{D}$.

The operation count is detailed in the following flowchart of Algorithm A.

Flowchart of operation count for computing $A^{(k-s)}$, s = 1 or 2



The comparison count is much less for Algorithm A than for the complete pivoting scheme, and in practice this fact has had a much larger impact than originally anticipated, sometimes cutting the execution time by about 40%. In the complete pivoting scheme, at each stage of the algorithm the largest element in the current submatrix is determined and the comparison count is bounded by $n^3/6$ and $n^2/2 + \frac{n}{3}$ (see [2]), while in Algorithm A at most 2 columns are searched and the comparison count is at most $n^2 - 1$.

The table given below gives upper bounds for the number of operations required for solving Ax=b by Algorithm A.

Table of Operation Count

	Multiplications	Divisions	Additions	Comparisons
Decision Phase	$4(n-1)$	0	0	n^2-1
Decomposition Phase	$\frac{n^3}{6} + \frac{3n^2}{4} - \frac{7n}{6}$	$\frac{n(n-1)}{2}$	$\frac{n^3}{6} + \frac{n^2}{4} - \frac{2n}{3}$	0
Back and forward solving	n^2	n	n^2-n	0
Total	$\frac{n^3}{6} + \frac{7n^2}{4} - \frac{7n}{6}$	$\frac{3n^2}{2} - \frac{n}{2}$	$\frac{n^3}{6} + \frac{5n^2}{4} - \frac{5n}{3}$	n^2-1

3. Variations of the Algorithm

3.1 Estimating $\mu^{(k)}$. For small n, we can construct examples for which the element growth bound of $(2.57)^{n-1}\mu$ for Algorithm A of section 2 is attained. However, we have been unable to construct an example for arbitrary n which reaches $(2.57)^{n-1}\mu$. Furthermore, as with Gaussian elimination with partial pivoting, large growth does not seem to occur in practice. Nevertheless, one would like to have a quick method for obtaining an estimate of $\mu^{(k)}$ so that whenever the element growth is excessive, a switch can be made to the slower executing complete pivoting scheme of Bunch and Parlett [4], for which the element growth is bounded by $nf(n)c(\alpha)h(\alpha,n)$ where $f(n) = \prod_{k=2}^{n}(k^{1/(k-1)})^{1/2}$ which grows slowly like $n^{\frac{1}{4}\log(n)}$, and

$c(\alpha)h(\alpha,n) \leq 3\ n$ for $\alpha_o = (1+\sqrt{17})/8$.

Businger [5] has presented an inexpensive algorithm for monitoring the growth in Gaussian elimination with partial pivoting. Because of the symmetry of our decomposition, Businger's idea is very satisfying when applied to Algorithm A.

According to (2.1), (2.2), and (2.3) when a 1 x 1 pivot is used to find $A^{(k-1)}$,

$$\mu^{(k-1)} \leq \mu^{(k)} + \beta^{(k)} ,$$

where
$$\beta^{(k)} = \begin{cases} \lambda^{(k)}/\alpha & \text{if } A^{(k-1)} \text{ is formed at step (2)} \\ \sigma^{(k)}/\alpha & \text{if } A^{(k-1)} \text{ is formed at steps (4) or (5).} \end{cases}$$

According to (2.6) when a 2 x 2 pivot is used to find $A^{(k-2)}$,

$$\mu^{(k-2)} \leq \mu^{(k)} + \beta^{(k)} + \beta^{(k-1)}$$

where $\beta^{(k)} = \beta^{(k-1)} = \sigma^{(k)}/(1-\alpha)$.

Therefore

$$\mu^{(k)} \leq \mu + \sum_{j=k+1}^{n} \beta^{(j)} .$$

Thus only the $n^2/2$ comparisons to determine μ, and k divisions and k additions are needed to determine a decent estimate to $\mu^{(k)}$.

We suggest that whenever

$$\mu^{(k)} \geq 13\, n$$

the complete pivoting strategy of Bunch and Parlett [4] should be used.

3.2 Accumulating inner products: Algorithm B.

In this section we describe a reformulation of Algorithm A which permits the accumulation of inner products in multiple precision and which would probably be more suitable for electronic hand calculators. The method, called Algorithm B, relates to Algorithm A in the same way that the Crout-Doolittle algorithm for solving a system of equations with a general matrix relates to Gaussian elimination with partial pivoting. (see [9]). Algorithm B was motivated by Aasen's [1] description of Parlett and Reid's [8] algorithm.

In Algorithm B, for a given symmetric matrix A, the matrices P,S,D, and M are computed such that

$$PAP^T = MS \tag{3.1}$$

and

$$S = DM^T \tag{3.2}$$

where P is a permutation matrix, M is a unit lower triangular matrix, and D is a block diagonal matrix where each block is either of order 1 or 2. Whenever $D_{k+1,k} = 0$, $M_{k-1,k}=0$. Because of the structure of M and D, the matrix S is an upper Hessenberg matrix with $S_{k+1,k}= 0$ whenever $D_{k+1,k}= 0$. In the parlance of section 2, a 1 x 1 pivot corresponds to $S_{k+1,k}= 0$. The decision to use a 1 x 1 pivot is based on the same criteria as in Algorithm A.

Algorithm B is developed by equating both sides of equations (3.1) and (3.2). At the k^{th} step, the first k-1 columns of M and P and the first k-1 rows of S, D, and P have been determined. It is assumed that the bottom (n-k+1) submatrix of P is the identity matrix. Equating the k^{th} column of (3.2) yields $S_{i,k}$ for $i < k$ and equating the k^{th} columns of (3.1) yields $S_{k,k}$. If a 1 x 1 pivot is used, then from (3.1)$M_{i,k}$ for $i > k$ can be obtained and by (3.2) $D_k = S_{k,k}$. On the other hand, if a 2 x 2 pivot is used, then equation (3.2) gives $S_{i,k+1}$ for $i < k$ and equation (3.1) gives $S_{k+1,k+1}$ and $S_{k+1,k}$. The remaining elements of the k^{th} and $(k+1)^{st}$ columns of M are found by solving (n-k-1) systems of order 2 obtained by equating the k^{th} and $(k+1)^{st}$ columns of both members of (3.1). The k^{th} and $(k+1)^{st}$ rows of D are immediate from (3.2).

More precisely, at the k^{th} stage one proceeds as follows:

(1) Compute S_{ik}: $= D_{i,i}M_{k,i} + D_{i,i-1}M_{k,i-1} + D_{i,i+1}M_{k,i+1}$

for $i < k$.

(Note either $D_{i,i-1}$ or $D_{i,i+1}$ is zero or both are zero.)

Compute Q_i: $= A_{ik} - \sum_{j=1}^{k-1} M_{ij}S_{jk}$. for $i \geq k$.

(2) Let $\lambda := |Q_j| := \max_{k<i\leq n} |Q_i|$.

If $|Q_k| \geq \alpha\lambda$ where $0 < \alpha < 1$,

set $M_{ik} := A_i/Q_k$ for $i > k$,

$D_{kk} := S_{kk} := Q_k$,

$D_{k+1,k} := D_{k,k+1} := S_{K=1,k} := 0$,

increment k by 1 and return to (1).

As in Algorithm A, a good value of α is $(1+\sqrt{17})/8$.

(3) Interchange the $k+1^{st}$ and j^{th} rows of M, then interchange the $k+1st$ and j^{th} rows and columns of A, and interchange Q_j and Q_{k+1}.

Compute $S_{i,k+1} = D_{ii}M_{ji} + D_{i,i-1}M_{j,i-1} + D_{i,i+1}M_{j,i+1}$

for $i < k$,

and $R_i := A_{ij} - \sum_{m=1}^{k-1} M_{im}S_{m,k+1}$ for $i > k$.

(4) If $\sigma|Q_k| \geq \alpha\lambda^2$, then set

$D_{kk} := S_{kk} := Q_k$,

$D_{k+1,k} := D_{k,k+1} := S_{k+1,k} = 0$,

and $M_{ik} := Q_i/Q_k$ for $i > k$,

and increment k by 1 and return to 1.

(5) Let $\sigma = \max(\lambda, \max_{k+1<i\leq n} |R_i|)$.

If $|R_k| > \alpha\ \sigma$, then interchange the k^{th} and $(k+1)^{st}$ rows of M and interchange the k^{th} and $(k+1)^{st}$ rows and columns of A and set

$$D_{kk} := S_{kk} := R_{k+1},$$

$$D_{k+1,k} := D_{k,k+1} := S_{k+1,k} := 0,$$

$$M_{i,k} := R_i/D_{kk} \quad \text{for } i > k+1,$$

and $M_{k+1,k} := Q_{k+1}/D_{kk}$,

and increment k by 1 and return to step (1).

(6) Set $D_{kk} := S_{kk} := Q_k$,

$$S_{k+1,k} := S_{k,k+1} := D_{k,k+1} := D_{k+1,k} := Q_{k+1} ,$$

$$D_{k+1,k+1} := S_{k+1,k+1} := R_{k+1} ,$$

$$M_{ik} := (Q_iR_{k+1}-R_iQ_{k+1})/(Q_kR_{k+1}-Q_{k+1}^2) \quad \text{for } i > k+1$$

$$M_{i,k+1} := (R_iQ_k-Q_iQ_{k+1})/(Q_kR_{k+1}-Q_{k+1}^2) \quad \text{for } i > k+1$$

$$M_{k+1,k} := D_{k+2,k+1} := 0 ,$$

increment k by 2 and return to (1).

If at the $(k-1)^{st}$ stage of the algorithm, execution terminated at either step 4 or 5, the quantities $S_{i,k}$ for $i < k$ and Q_i for $i \geq k$ at the k^{th} stage have been computed except for one multiplication and one addition. If step (1) is modified to reflect this fact, then Algorithm B requires the same number of multiplications as Algorithm A.

In practice the matrices M,A,S, and D and the vectors Q and R may be stored in an nxn array.

As before, an optimal value of α is $\alpha_0 = (1+\sqrt{17})/8$ and for α_0,

$$|D_{ij}| \leq (2.57)^{n-1} \max_{1\leq i,j\leq n} |A_{ij}| \quad .$$

3.3 Other strategies. Other partial pivoting strategies, similar to Algorithm A, exist, including Algorithms C and D given below.

In Algorithm C, $A^{(k-s)}$ is determined as follows:

(1) Determine $\mu_1{}^{(k)} = |A_{pp}^{(k)}| = \max_{1\leq i\leq k} |A_{ii}^{(k)}|$

and permute the first and p^{th} rows and columns of $A^{(k)}$ so that $|A_{ii}^{(k)}| = \mu_1{}^{(k)}$.

(2) Determine $\lambda^{(k)} = |A_{j1}^{(k)}| = \max_{2\leq i\leq k} |A_{i1}^{(k)}|$.

(3) If $\mu_1{}^{(k)} \geq \alpha\, \lambda^{(k)}$, use $A_{11}^{(k)}$ as a 1 x 1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1). As before a good value for α is $(1+\sqrt{17})/8$.

(4) Determine $\sigma^{(k)} = \max_{\substack{2\leq m\leq k \\ m\neq j}} |A_{m,j}^{(k)}|$

(5) If $\alpha\, \lambda^{(k)2} \leq |A_{11}^{(k)}|\ \sigma^{(k)}$, use $A_{11}^{(k)}$ as a 1 x 1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1).

(6) Interchange the second and j^{th} rows and columns of $A^{(k)}$ so that $|A_{21}^{(k)}| = \lambda^{(k)}$ and perform a 2 x 2 pivot to obtain $A^{(k-2)}$, decrement k by 2 and return to (1).

Because the maximum element of a positive definite matrix is on the diagonal, when Algorithm C is applied to a positive definite matrix A,

one obtains the decomposition

$$PAP^t = MDM^t$$

with $|M_{ij}| \leq 1$. For some applications this is very desirable. Unfortunately, on most problems, Algorithm C is more costly than Algorithm A because at each stage the diagonal is searched and extra interchanges might be required. In Algorithm A between $n^2/2 + O(n)$ and $n^2 + O(n)$ comparisons are needed to determine the pivot strategy while in Algorithm C between $3n^2/4 + O(n)$ and $3n^2/2 + O(n)$ comparisons are needed to determine the pivot strategy. The bound on element growth in $A^{(k)}$ for Algorithm C is the same as for Algorithm A.

In Algorithm D, $A^{(k-s)}$ is determined as follows:

(1) Determine $\lambda^{(k)} = |A_{j1}^{(k)}| = \max_{2 \leq i \leq k} |A_{i1}^{(k)}|$.

(2) If $A_{11}^{(k)} \geq \alpha\lambda^{(k)}$, use $A_{11}^{(k)}$ as a 1 x 1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1). Below we shall show that a good value of α is about 0.525.

(3) Determine $\sigma^{(k)} = \max_{2 \leq m \leq k} |A_{m,j}^{(k)}|$.

(4) If $\alpha\,\lambda^{(k)^2} \leq |A_{11}^{(k)}|\,\sigma^{(k)}$, then use $A_{11}^{(k)}$ as a 1 x 1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1).

(5) Interchange the second and j^{th} rows and columns of $A^{(k)}$ so that $|A_{21}^{(k)}| = \lambda^{(k)}$, perform a 2 x 2 pivot to obtain $A^{(k-2)}$, decrement k by 2 and return to (1).

Whenever a 1 x 1 pivot is used in Algorithm D, no interchanges are performed, which means less bookkeeping, fewer references to memory in general, and fewer opportunities to interfere with the structure of the system. In particular, the algorithm is quite amenable to tridiagonal systems.

The disadvantage of Algorithm D is a larger bound in the element growth in the matrix. As in § 2, let $\mu^{(k)} = \max_{1\le i,j\le k} |A_{ij}^{(k)}|$. As in Algorithm A, whenever a 1 x 1 pivot is used,

$$\mu^{(k-1)} \le \mu^{(k)}(1 + 1/\alpha) .$$

When a 2 x 2 pivot is used,

$$|A_{11}^{(k)}|\ |A_{22}^{(k)}| \le |A_{11}^{(k)}|\ \sigma^{(k)} < \alpha\, \lambda^{(k)2} ,$$

so $\nu = |A_{11}^{(k)} A_{22}^{(k)} - A_{21}^{(k)2}| > \lambda^{(k)2}(1-\alpha)$, which is a slightly smaller bound on ν than in Algorithm A.

Because $|A_{22}^{(k)}| < \sigma^{(k)}$ and $|A_{11}^{(k)}|\ \sigma^{(k)} < \alpha\, \lambda^{(k)2}$, equation (2.4) implies

$$\mu^{(k-2)} \le [1+(3+\alpha)/(1-\alpha)]\ \mu^{(k)}.$$

Thus $\mu^{(k)} \le \max\left\{(1+1/\alpha)^{n-k}, [1+(3+\alpha)/(1-\alpha)]^{(n-k)/2}\right\}\mu$,

which is minimized when

$$(1+1/\alpha)^2 = 1 + (3+\alpha)/(1-\alpha) .$$

This occurs when α is approximately 0.525, giving a bound of $(2.92)^{n-1}\mu$, which is larger than in Algorithm A.

4.1 Band Matrices

Many of the problems in numerical linear algebra with symmetric indefinite matrices involve band matrices. A band matrix A is said to have bandwidth m if $A_{ij} = 0$ for $|i-j| > m$. When A is band one would like to use an algorithm, like Gaussian elimination with partial pivoting, which takes advantage of the band structure of the matrix to increase the efficiency of the algorithm.

Unfortunately, except for m=1 and m=2, none of the algorithms outlined in sections 2 and 3 guarantee the preservation of the band structure of the matrix. The row and column interchanges used to guarantee stability destroy the band structure of the system.

Algorithm D does the least damage of all the algorithms A - D, since interchanges only occur when a 2 x 2 pivot is used, and hence only in this case is the bandwidth increased. Let m_k be the bandwidth of the matrix $A^{(k)}$ generated by Algorithm D. When a 1 x 1 pivot is used,

$$m_{k-1} = \begin{cases} m_k - 1 & \text{if } m_k > m \\ m_k & \text{otherwise} \end{cases} .$$

When a 2 x 2 pivot is used, $m_{k-2} = \max(m_k - 2, m, j+m-2)$ where the j^{th} and 2^{nd} columns are interchanged before the creation of $A^{(k-2)}$. Since $j \leq m_k$, one is assured that $m_{k-2} \leq m_k + m - 2$ and $m_k \leq m + \frac{n-k}{2}(m-2)$. For $m > 2$, one must concede that the band structure might be ruined.

In section 4.2 we discuss the tridiagonal case (m = 1) and in section 4.3 we present an algorithm for the 5 diagonal case (m = 2).

4.2 Tridiagonal Matrices

Let T by a symmetric, tridiagonal matrix, i.e. $T_{ij} = 0$ for $|i-j|>1$. Of the many algorithms that have been proposed to solve $Tx = b$, Gaussian elimination with partial pivoting has proved the least time-consuming. However, Gaussian elimination with partial pivoting does not preserve symmetry. In [3] Bunch has proposed a symmetry preserving algorithm which can be used to determine the inertia of T as well as solve a system of equations. Like those given in sections 2 and 3, the algorithm finds the MDM^t decomposition of (1.1) by generating a sequence of tridiagonal matrices $T^{(k)}$ of order k. We show the first step which is typical:

Let α be a fixed number such that $0<\alpha<1$.

1) If $|T_{11}| \geq \alpha T_{21}^2$, then use a 1 x 1 pivot to generate $T^{(n-1)}$.

2) If $|T_{11}| < \alpha T_{21}^2$, then use a 2 x 2 pivot to generate $T^{(n-2)}$.

Bunch [3] shows that the bound on element growth is minimized when $\alpha = (\sqrt{5}-1)/(2\mu)$ where $\mu = \max_{1\leq i,j\leq n} |T_{ij}|$. With this value of α,

$$\max_{1\leq i,j\leq k} |T_{ij}^{(k)}| \leq \frac{(3+\sqrt{5})}{2}\mu$$

Table 4.1 gives the operation counts and storage requirements for Bunch's algorithm [3] and Gaussian elimination with partial pivoting. When storage is crucial, Bunch's algorithm [3] is preferable to Gaussian elimination with partial pivoting.

Table 4.1: Operation Count: Tridiagonal Case.

	Bunch's Original Algorithm [3]		Modified Algorithm		Gaussian Elimination with Partial Pivoting	
	Decomposition Only	Solving Tx=b	Decomposition Only	Solving Tx=b	Decomposition Only	Solving Tx=b
Multiplications	$3\frac{1}{2}n+\frac{1}{2}p$	$8\frac{1}{2}n-\frac{3}{2}p$	$3\frac{1}{2}n-\frac{1}{2}p$	$7\frac{1}{2}n-\frac{3}{2}p$	3n	7n
Additions	n	4n-p	n	4n-p	2n	5n
Comparisons	$2\frac{1}{2}n+\frac{1}{2}p$	$2\frac{1}{2}n+\frac{1}{2}p$	3n+p	3n+p	n	n
Storage required	3n	4n	3n	4n	5n	6n

p represents the number of 1 x 1 pivots

For certain huge problems, where the whole matrix cannot fit into storage, and for applications where it is not always necessary to have the complete decomposition, Bunch's algorithm has the unfortunate aspect that to determine α the whole matrix must be examined to find μ. This problem can be remedied by changing the test in step (1) to:

1) If $\max(|A_{21}|, |A_{22}|, |A_{32}|) \times |A_{11}| \geq \alpha A_{21}^2$, then use a 1 x 1 pivot. Here α is simply $(\frac{3+\sqrt{5}}{2})$.

The bound on element growth with this modification is the same, but the decomposition now requires 4n+p multiplications and $\frac{3}{2}n+\frac{3}{2}p$ comparisons.

Bunch's original algorithm can be modified slightly to obtain an operation count closer to that of Gaussian elimination when solving linear equations. The modification involves realizing that one need not construct the MDM^t decomposition explicitly but only that part of the decomposition which is useful in solving linear equations.

To solve Ax = b one solves Mc = b for c, Dy = c for y and $M^t x = y$ x. Let us assume that the first block of D is 2 x 2 and hence

$$y_1 = (D_{11}c_1 - D_{21}c_2)/\eta \quad \text{and} \quad y_2 = (D_{22}c_2 - D_{21}c_1)/\eta$$

where $\eta = D_{22}D_{11} - D_{21}^2$. Since η is also needed during the formation of D and M, it is usually available. One may also write

$$y_1 = (\beta c_1 - c_2)/\delta \text{ and } y_2 = (\gamma c_2 - c_1)/\delta$$

where $\beta = D_{11}/D_{21}$, $\gamma = D_{22}/D_{21}$ and $\delta = D_{22}D_{11}/D_{21} - D_{21} = D_{22}\beta - D_{21}$.

If β,γ, and δ have been constructed in the decomposition phase and saved in place of D_{11}, D_{21}, and D_{22}, then two multiplications are eliminated. Since

$$M_{31} = -T_{32}/\delta \quad \text{and} \quad M_{32} = \beta M_{31},$$

the quantities β and δ can be used in the decomposition phase, and, in fact, constructing β,γ, and δ does not add to the multiplication count when forming M and D when a 2 x 2 is used.

If a 1 x 1 is performed whenever $|T_{11}| \geq |T_{21}|$, or whenever $|\beta| \geq \alpha |T_{21}|$, a multiplication is eliminated for each 1 x 1 pivot in the decomposition phase and the result is equivalent to that obtained with Bunch's original scheme. Of course, one must still perform the extra comparison, $|T_{11}| > |T_{21}|$, which avoids problems in the formation of β.

Columns 3 and 4 of Table 4.1 give the operation count for this algorithm.

4.3 Five-diagonal Matrices.

In this section we consider two methods for a symmetric indefinite five-diagonal matrix F, i.e. $F_{ij} = 0$ for $|i-j| > 2$. Such a matrix arises during the solution of partial differential equations with periodic boundary conditions. As in the case of tridiagonal matrices, Gaussian elimination with partial pivoting is still the least time-consuming stable algorithm for solving $Fx = b$, but it destroys symmetry. In this section we describe two symmetry-preserving algorithms, E and F, which, for an irreducible matrix F, determine matrices P, M, and D such that

(4.1) $$PFP^t = MDM^t$$

as in (1.1). Here $M_{ij} = 0$ for $i > j + 3$. With decomposition (4.1) one can solve $Fx = b$ with less storage than Gaussian elimination with partial pivoting but with a slightly higher operation count.

The algorithms follow the ideas used in sections 2 and 3 and generate a sequence of five diagonal matrics $F^{(k)}$ of order k. They were designed so that the bound on the element growth of $F^{(k)}$ is independent of k and the operation count is kept as low as possible.

Both algorithms have the same bound on element growth. The bound on the operation count for Algorithm E is slightly higher than that of Algorithm F, but in Algorithm E the probability of attaining the bound is less.

The first step of each algorithm is typical.

Algorithm E:

(1) If $|F_{21}| \geq |F_{31}|$, then

let $\sigma = \max (|F_{21}| , |F_{32}| , |F_{42}|)$.

(a) If $\sigma|F_{11}| \geq \alpha F_{21}^2$, generate $F^{(n-1)}$ using a 1 x 1 pivot.

(b) If $|F_{22}| \geq \sigma$, then interchange the first and second rows and columns of F and perform a 1 x 1 pivot on the new F to generate $F^{(n-1)}$.

(c) Use a 2 x 2 pivot to generate $F^{(n-2)}$.

(2) If $|F_{21}| < |F_{31}|$, then

let $\sigma = \max_{2 \leq i \leq 5} |F_{i3}|$.

(a) If $\sigma |F_{11}| \geq \alpha F_{i3}^2$, then perform a 1 x 1 pivot to generate $F^{(n-1)}$.

(b) Interchange the second and third rows and columns of F and perform a 2 x 2 pivot step to generate $F^{(n-2)}$.

Algorithm F:

(1) If $|F_{21}| \geq |F_{31}|$, then

(a) if $|F_{11}| \geq \alpha |F_{21}|$, then generate $F^{(n-1)}$ using a 1 x 1 pivot.

(b) Let $\sigma = \max (|F_{21}|, |F_{32}|, |F_{42}|)$.

If $|F_{22}| \geq \sigma$, then interchange the first and second rows and columns of F and perform a 1 x 1 pivot on the new F.

(c) If $\sigma |F_{11}| \geq \alpha F_{21}^2$, generate $F^{(n-1)}$ using a 1 x 1 pivot.

(d) Use a 2 x 2 pivot to generate $F^{(n-2)}$.

(2) If $|F_{21}| < |F_{31}|$ then do the same as (2) in Algorithm E.

Step (1b) in each algorithm was included to ensure that the bound on $|F_{22}^{(k)}|$ would be independent of k. In Algorithm F step (1a) was included so that the bound on $|F_{11}^{(k)}|$ would be independent of k.

In both algorithms, $F^{(k)}$ is 5-diagonal for all k, and the lower submatrix of $F^{(k)}$ of order (k-3) is that of F. Thus if $\max_{1\le i,j\le n} |F_{ij}| \le \mu$, then $|F^{(k)}_{ij}| \le \mu$ for $i > 3$ and $j > 3$.

A complete analysis of the element growth for Algorithm E is given below. The main results are also valid for Algorithm F and follow similarly.

It is easiest to bound the element growth of the matrices in Algorithm E by bounding the elements in the third row of $F^{(k)}$, using these results to bound $|F^{(k)}_{22}|$ and $|F^{(k)}_{21}|$, and in turn using these bounds to obtain a bound on $|F^{(k)}_{11}|$.

The third row:

In steps 1a, 1c, and 2a, the third row is not affected so $|F^{(k-1)}_{3i}| \le \mu$.

In step 1b , $|F^{(k)}_{22}| \ge |F^{(k)}_{i2}|$ for i = 1,2,3,4 so that

$$|F^{(k-1)}_{31}| = |-F^{(k)}_{42}\ F^{(k)}_{21} / F^{(k)}_{22}| \le |F^{(k)}_{42}| \le \mu$$

$$|F^{(k-1)}_{32}| = |F^{(k)}_{43} - F^{(k)}_{32}\ F^{(k)}_{42} / F^{(k)}_{22}| \le |F^{(k)}_{43}| + |F^{(k)}_{42}| \le 2\mu$$

$$|F^{(k-1)}_{33}| = |F^{(k)}_{44} - F^{(k)}_{42}\ F^{(k)}_{42} / F^{(k)}_{22}| \le |F^{(k)}_{44}| + |F^{(k)}_{42}| \le 2\mu$$

In step 2b, $\max_{2\le i\le 5} |F_{11}^{(k)} F_{3i}^{(k)}| < \alpha F_{31}^{(k)2}$ so that

$$|F_{31}^{(k-2)}| = \frac{|-F_{53}^{(k)} (F_{31}^{(k)} F_{21}^{(k)} + F_{11}^{(k)} F_{32}^{(k)})|}{|F_{33}^{(k)} F_{11}^{(k)} - F_{31}^{(k)2}|}$$

$$\le \frac{|F_{53}^{(k)}| F_{31}^{(k)2} (1+\alpha)}{F_{31}^{(k)2}(1-\alpha)} = \mu\frac{(1+\alpha)}{(1-\alpha)}$$

and for j=2,3:

$$|F_{3j}^{(k-2)}| = \left| F_{5,2+j}^{(k)} - \frac{F_{53}^{(k)} F_{11}^{(k)} F_{2+j,3}^{(k)}}{(F_{33}^{(k)} F_{11}^{(k)} - F_{31}^{(k)2})} \right|$$

$$\le |F_{5,2+j}^{(k)}| + |F_{53}^{(k)}| \frac{\alpha}{1-\alpha} \le \mu\left\{1 + \frac{\alpha}{1-\alpha}\right\} = \mu\left(\frac{1}{1-\alpha}\right)$$

Since $1 < \frac{1+\alpha}{1-\alpha}$ for $0<\alpha<1$, we know

$$|F_{31}^{(k)}| \le \mu\left(\frac{1+\alpha}{1-\alpha}\right) \text{ for all } k.$$

Assuming $1/2\le\alpha<1$, one is assured that $|F_{32}^{(k)}| \le \mu\left(\frac{1}{1-\alpha}\right)$ and $|F_{33}^{(k)}| \le \mu\left(\frac{1}{1-\alpha}\right)$. A full analysis for the case $0<\alpha\le 1/2$ leads to the conclusion that the best value of α is 1/2. Thus one might as well assume $1/2\le\alpha<1$.

The second row:

In step 1a,

$$|F_{11}^{(k)}| \max (|F_{32}^{(k)}|, |F_{42}^{(k)}|\,|F_{21}^{(k)}|) \geq \alpha F_{21}^{(k)2} ,$$

which implies $|F_{11}^{(k)}| \geq \alpha \; |F_{21}^{(k)}|$

and $|F_{11}^{(k)}| \geq \alpha |F_{31}^{(k)}|$.

Thus, in 1a, for $j = 1,2$

$$|F_{2j}^{(k-1)}| = |F_{3,j+1}^{(k)} - F_{31}^{(k)} F_{1+j,1}^{(k)} / F_{11}^{(k)}|$$

$$\leq |F_{3,j+1}^{(k)}| + \frac{1}{\alpha} \max (|F_{31}^{(k)}|, |F_{42}^{(k)}|, |F_{32}^{(k)}|)$$

$$\leq \mu \left[\frac{1}{1-\alpha} + \frac{1}{\alpha} \left(\frac{1+\alpha}{1-\alpha} \right) \right].$$

If 1b had been deleted and the definition of σ in 1a changed to $\sigma = \max_{1 \leq i \leq 4} (|F_{2i}^{(k)}|)$

then the bound on $|F_{22}^{(k-1)}|$ would have been

$$|F_{22}^{(k-1)}| \leq \frac{1}{1-\alpha} \mu + \frac{1}{\alpha} \max \left(|F_{22}^{(k)}| , \frac{1+\alpha}{1-\alpha} \right),$$

that is, dependent on k.

In step 1b,

$$|F_{22}^{(k-1)}| = |F_{33}^{(k)} - F_{32}^{(k)2} / F_{22}^{(k)}| \leq |F_{33}^{(k)}| + |F_{32}^{(k)}| \leq \mu \left(\frac{2}{1-\alpha} \right)$$

and

$$|F_{21}^{(k-1)}| = |F_{31}^{(k)} - F_{21}^{(k)} F_{32}^{(k)} / F_{22}^{(k)}| \leq |F_{31}^{(k)}| + |F_{32}^{(k)}| \leq \mu \left(\frac{2+\alpha}{1-\alpha} \right).$$

In step 1c, when a 2 x 2 pivot is used, $|F_{11}^{(k)}| \max_i |F_{2i}| < \alpha F_{21}^{2}$

so $$|F_{22}^{(k-2)}| = \left| F_{44}^{(k)} - \frac{F_{42}^{(k)} F_{11}^{(k)} F_{42}^{(k)}}{F_{11}^{(k)} F_{22}^{(k)} - F_{21}^{(k)2}} \right|$$

$$< |F_{44}^{(k)}| + \frac{|F_{42}^{(k)}| F_{21}^{(k)2} \alpha}{F_{21}^{(k)2}(1-\alpha)} < \mu \left(1 + \frac{\alpha}{1-\alpha}\right) = \frac{\mu}{1-\alpha}$$

and $$|F_{21}^{(k-2)}| = \left| F_{43}^{(k)} - \frac{(F_{42}^{(k)} F_{21}^{(k)} F_{31}^{(k)} + F_{42}^{(k)} F_{11}^{(k)} F_{32}^{(k)})}{F_{11}^{(k)} F_{22}^{(k)} - F_{21}^{(k)2}} \right|$$

$$< |F_{43}^{(k)}| + \frac{|F_{42}^{(k)}| F_{21}^{(k)2}(1+\alpha)}{F_{21}^{(k)2}(1-\alpha)} \leq \mu \left[1 + \frac{(1+\alpha)}{(1-\alpha)}\right] = \mu\,(2/(1-\alpha)).$$

In 2a, $|F_{11}^{(k)}| \max_{2\leq i\leq 5} |F_{3i}^{(k)}| \leq |F_{31}^{(k)}|$,

so that for j=1,2

$$|F_{2j}^{(k-1)}| \leq |F_{3,j+1}^{(k)}| + \frac{1}{\alpha} \max_{2\leq i\leq 5} |F_{3i}^{(k)}| \leq \mu \left[\frac{1}{(1-\alpha)} + \frac{1}{\alpha}\left(\frac{1}{(1-\alpha)}\right)\right] = \mu \left|\frac{(1+1/\alpha)}{1-\alpha}\right|$$

The case of 2b is exactly that of 1c with the subscripts '2' and '3' interchanged. One can easily show that for step 2b,

$$|F_{22}^{(k-2)}| \leq \mu/(1-\alpha) \quad \text{and} \quad |F_{21}^{(k-2)}| \leq \mu(2/(1-\alpha)).$$

Thus, assuming $\frac{1}{2} \leq \alpha < 1$,

$$|F_{22}^{(k)}| \leq \mu\,(2+1/\alpha)/(1-\alpha) \quad \text{and} \quad |F_{21}^{(k)}| \leq \mu(2+1/\alpha)/(1-\alpha)$$

for all k.

In order to bound $|F_{11}^{(k)}|$, we do the following:

In step 1a,

$$(4.2)\quad |F_{11}^{(k-1)}| = |F_{22}^{(k)} - F_{21}^{(k)2}/F_{11}^{(k)}| \le |F_{22}^{(k)}| + \frac{1}{\alpha}\max(|F_{12}^{(k)}|, |F_{32}^{(k)}|, |F_{42}^{(k)}|)$$

$$\le \mu\,(1/\alpha + 1)(2 + 1/\alpha)/(1-\alpha).$$

In step 1b,

$$|F_{11}^{(k)}|\max(|F_{12}^{(k)}|, |F_{32}^{(k)}|, |F_{42}^{(k)}|) \le \alpha\, F_{12}^{(k)2}, \quad \text{which implies}$$

$$|F_{11}^{(k)}| \le \alpha\, |F_{12}^{(k)}| \qquad \text{so that}$$

$$(4.3)\quad |F_{11}^{(k-1)}| = |F_{11}^{(k)} - F_{21}^{(k)2}/F_{22}^{(k)}| \le (1+\alpha)|F_{21}^{(k)}| \le \mu(1+\alpha)(2+1/\alpha)/(1-\alpha).$$

In step 1c,

$$|F_{11}^{(k-2)}| = \left| F_{33}^{(k)} - \left[\frac{(F_{31}^{(k)}F_{22}^{(k)} - F_{32}^{(k)}F_{21}^{(k)})F_{31}^{(k)} + (F_{32}^{(k)}F_{11}^{(k)} - F_{31}^{(k)}F_{21}^{(k)})F_{32}^{(k)}}{F_{22}^{(k)}F_{11}^{(k)} - F_{21}^{(k)2}} \right] \right|$$

$$\le |F_{33}^{(k)}| + \frac{(|F_{31}^{(k)}| + |F_{32}^{(k)}| + \alpha|F_{32}^{(k)}| + |F_{32}^{(k)}|)\, F_{21}^{(k)2}}{F_{21}^{(k)2}(1-\alpha)}$$

$$(4.4)\qquad \le \mu\left[\frac{1}{(1-\alpha)} + \frac{\frac{(1+\alpha)}{(1-\alpha)} + \frac{(2+\alpha)}{(1-\alpha)}}{(1-\alpha)} \right] = \mu\,(4+2\alpha)/(1-\alpha)^2.$$

In step 2a,

$$(4.5)\quad |F_{11}^{(k-1)}| \le |F_{22}^{(k)}| + \frac{1}{\alpha}\max_{2\le i\le 5}|F_{3i}^{(k)}| \le \mu\,(2+2/\alpha)/(1-\alpha),$$

and in step 2b,

$$|F_{11}^{(k-2)}| = \left| F_{22}^{(k)} - \frac{(F_{21}^{(k)}F_{33}^{(k)} - F_{32}^{(k)}F_{31}^{(k)})F_{21}^{(k)} + (F_{32}^{(k)}F_{11}^{(k)} - F_{21}^{(k)}F_{31}^{(k)})F_{32}^{(k)}}{F_{33}^{(k)}F_{11}^{(k)} - F_{31}^{(k)^2}} \right|$$

$$\leq |F_{22}^{(k)}| + \frac{(|F_{33}^{(k)}| + |F_{32}^{(k)}| + \alpha|F_{32}^{(k)}| + |F_{32}^{(k)}|)F_{31}^{(k)^2}}{(1-\alpha)F_{31}^{(k)^2}}$$

$$(4.6) \qquad \leq \mu(2 + 1/\alpha) + (3+\alpha)/(1-\alpha))/(1-\alpha) = \mu(-\alpha^2 + 4\alpha + 1)/(\alpha(1-\alpha)^2).$$

Obviously, for $\alpha < 1$, the bound on $|F_{11}^{(k)}|$ of (4.2) is greater than those of (4.3) and (4.5). Whenever $\alpha^2 < 1/3$, the bound of (4.4) is less than that of (4.6), and within this range, the bound on $|F_{11}^{(k)}|$ is minimized when the formulae given in (4.2) and (4.6) are equal. This occurs when

$$(1+\alpha)(1-\alpha)(2\alpha+1) = (1 + 4\alpha - \alpha^2)$$

or

$$\alpha^3 + 5\alpha^2 - \alpha - 1 = 0,$$

which occurs when α is approximately 0.52524. For this value of α,

$$|F_{i,j}^{(k)}| \leq 23.88 \quad \mu \quad .$$

The operation count for each algorithm is largest when rows and columns are interchanged. The bound on the operation count is slightly higher for Algorithm E since more checking is done before the algorithm concedes that one must interchange rows and columns before performing a 1 x 1 pivot. But because of the extra checks, the bound will not be attained as often as it would be in Algorithm F.

In Table 4.2, p is the number of 1 x 1 pivots. If storage is crucial, Algorithm E or F should be used rather than Gaussian elimination with partial pivoting.

Table 4.2: Operation Count: Five diagonal Matrices

	Algorithm E		Algorithm F		Gaussian elimination with partial pivoting	
	Decomposition Only	Solving Fx=b	Decomposition Only	Solving Fx=b	Decomposition Only	Solving Fx=b
Multiplications:	$10n + 2p$	$19n$	$10n$	$19n-2p$	$10n$	$17n$
Additions:	$\frac{11}{2} n + \frac{1}{2} p$	$\frac{25}{2}n-\frac{1}{2}p$	$\frac{11}{2} n + \frac{1}{2} p$	$\frac{25}{2}n-\frac{1}{2}p$	$8n$	$14n$
Comparisons:	$\frac{5}{2} (n+p)$	$\frac{5}{2}(n+p)$	$3(n+p)$	$3(n+p)$	$2n$	$2n$
Storage:	$4n$	$5n$	$4n$	$5n$	$8n$	$9n$

For Algorithm F, the bounds on the multiplication, addition, and comparison count cannot all be attained simultaneously. The bounds on multiplication and addition are attained only if all 1 x 1 pivots are done in (1b) and all the 2 x 2's in (2b). In this case at most $\frac{5}{2} (n+p)$ comparisons can be done. The bound on the comparison count is attained only if all 1 x 1 pivots are done in (1c) and all the 2 x 2's in (1d). In this case 9n-2p additions and 15n-p multiplications are needed to solve a system of equations.

References

1. J. O. Aasen, "On the reduction of a symmetric matrix to tridiagonal form", BIT, 11 (1971), pp. 233-242.

2. J. R. Bunch, "Analysis of the diagonal pivoting method", SIAM Numerical Analysis, 8 (1971), pp. 656-680.

3. J. R. Bunch, "Partial pivoting strategies for symmetric matrices", SIAM Numerical Analysis, 11 (1974), pp. 521-528.

4. J. R. Bunch and B. N. Parlett, "Direct methods for solving symmetric indefinite systems of linear equations", SIAM Numerical Analysis 8 (1971), pp. 639-655.

5. P. A. Businger, "Monitoring the numerical stability of Gaussian elimination", Numerische Mathematik 16 (1971), pp. 360-361.

6. R. W. Cottle, "Manifestations of the Schur complement", Linear Algebra and its Applications, 8 (1974), pp. 189-211.

7. L. Mirsky, An Introduction to Linear Algebra, Clarendon Press, Oxford, 1955.

8. B. N. Parlett and J. K. Reid, "On the solution of a system of linear equations whose matrix is symmetric but not definite", BIT, 10 (1970), pp. 386-397.

9. J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.